

Claim 1 (Original): A method for building a time-sliced architecture in a spread spectrum system, comprising the steps of:

- (a) analyzing a set of applications, said analyzing including the steps of:
 - (i) extracting real time aspects from each application in said set of applications;
 - (ii) determining an optimal granularity based on said real time aspects; and
 - (iii) adjusting said optimal granularity based on a context switching overhead; and
- (b) building a specific time-sliced architecture to accommodate said range of applications based on said analyzing.

Claim 2 (Original): The method of claim 1, wherein said extracting includes the step of:
profiling fundamental processing elements in each application in said set of applications.

Claim 3 (Original): The method of claim 1, wherein said determining includes the step of:
determining a lowest level of granularity needed for each application in said set of applications.

Claim 4 (Original): The method of claim 1, wherein said adjusting includes the step of:
performing a sensitivity analysis.

Claim 5 (Original): The method of claim 4, wherein said performing includes the step of:
determining an optimal trade-off between said context switching overhead and said optimal granularity.

Claim 6 (Original): The method of claim 1, wherein said building includes the steps of:
determining a size for a data cache based on said extracting;
implementing a hierarchical caching structure in said data cache; and
applying said data cache in said specific time-sliced architecture.

logic code for determining an optimal trade-off between said context switching overhead and said optimal granularity.

logic code for determining a size for a data cache based on said extracting;
logic code for implementing a hierarchical caching structure in said data cache; and
logic code for applying said data cache in said specific time-sliced architecture.

a master control unit including a time slot table and a partial sums search table;
a data cache for receiving input data and for caching intermediate data; and
a plurality of finger processing elements, each element comprising:

a cache for receiving data from the data cache and for caching intermediate data,
a data selector connected to an output of the cache,
a despreader connected to an output of the data selector, and
a symbol integrator connected to an output of the despreader.

Claim 15 (New): The computer program product of claim 7, wherein the computer program product is independent of a communication protocol.

{W:\04303\100N155000\00601661.DOC {00000000-0000-0000-0000-000000000000}}